

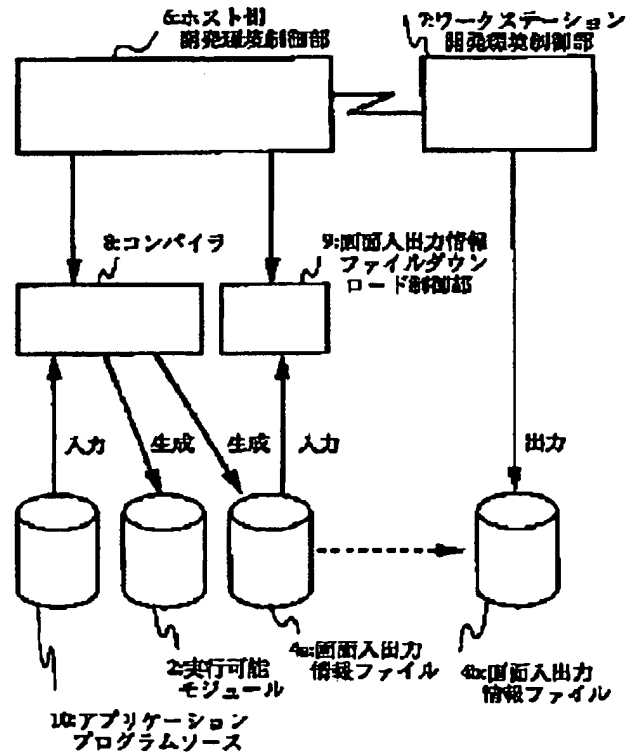
METHOD FOR IMPROVING TERMINAL OPERABILITY OF APPLICATION PROGRAM USE

Patent number: JP8329004
Publication date: 1996-12-13
Inventor: ITO MASAHIRO
Applicant: MITSUBISHI ELECTRIC CORP
Classification:
 - International: G06F15/00; G06F3/14
 - european:
Application number: JP19950134043 19950531
Priority number(s):

Abstract of JP8329004

PURPOSE: To improve the terminal operability of application program use by downloading an extended input/output information file, obtained by extending a screen input/output information file, from a host computer to a terminal and outputting or storing input/output data in a memory by referring to the extended input/output information file.

CONSTITUTION: The area of screen input/output information files 4a and 4b referred to at the time of input/output operation is expanded and additional information is added to text information regarding the application program of an application program source 10 to generate the extended input/output information file. A screen input/output information file downloading control part 10 downloads the extended input/output information file from the host terminal to the terminal. The terminal outputs or stores the input/output data in the memory by referring to the extended input/output information file. Consequently, the terminal operability of application program use is improved.



Data supplied from the esp@cenet database - Patent Abstracts of Japan

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-329004

(43) 公開日 平成8年(1996)12月13日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/00	3 1 0	9364-5L	G 0 6 F 15/00	3 1 0 S
3/14	3 1 0		3/14	3 1 0 C

審査請求 未請求 請求項の数2 O L (全 13 頁)

(21) 出願番号 特願平7-134043

(22) 出願日 平成7年(1995)5月31日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 伊藤 正裕

鎌倉市大船五丁目1番1号 三菱電機株式

会社情報システム研究所内

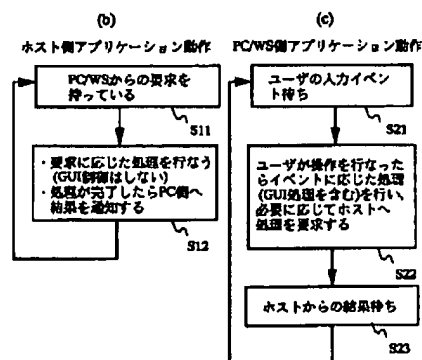
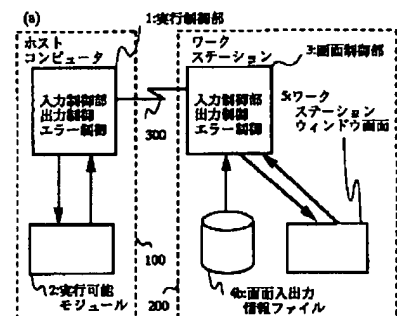
(74) 代理人 弁理士 宮田 金雄 (外3名)

(54) 【発明の名称】 アプリケーション・プログラム使用の端末操作性向上方法

(57) 【要約】

【目的】 既存のアプリケーション・プログラムを有効利用し、ソフトウェアの生産性と保守性を向上して、しかも端末での操作性を向上した端末操作性向上方法を得る。

【構成】 入出力を実行する際に参照する入出力情報ファイルの領域を拡張し、アプリケーション・プログラムに関係するテキスト情報に付加情報を付加して拡張入出力情報ファイルを作成するステップと、拡張入出力情報ファイルをホスト計算機から端末へダウンロードするステップと、端末では拡張画面制御部を設け、拡張画面制御部が拡張入出力情報ファイルを参照して入出力データを出力またはメモリへ記憶するステップとを備えた。



【特許請求の範囲】

【請求項1】 入出力を実行する際に参照する入出力情報ファイルの領域を拡張し、アプリケーション・プログラムに係るテキスト情報に付加情報を付加して拡張入出力情報ファイルを作成するステップと、上記拡張入出力情報ファイルをホスト計算機から端末へダウンロードするステップと、上記端末では拡張画面制御部を設け、上記拡張画面制御部が上記拡張入出力情報ファイルを参照して入出力データを出力またはメモリへ記憶するステップとを備えたアプリケーション・プログラム使用の端末操作性向上方法。

【請求項2】 拡張入出力情報ファイルは、テキスト情報を索引部とチェック情報を含む詳細情報とに分け、上記詳細情報部分のみに付加情報を追加することを特徴とする請求項1記載のアプリケーション・プログラム使用の端末操作性向上方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、ホスト計算機とワークステーション等で構成されるシステムで、アプリケーションプログラムを有効利用して、しかも端末操作性を向上した制御方法に関するものである。

【0002】

【従来の技術】 従来、この種の入出力制御方法として、特開平6-35855号公報に示された図25に示すようなものがあった。図において、ホスト計算機側にある1bは実行制御部で実行可能モジュールから通知された画面／ファイル／プリンタに対する入出力を制御するプログラム、2aは実行可能モジュール、4aは画面入出力情報ファイルで、1つの実行可能モジュールの画面を制御する情報を格納した情報ファイルである。また、端末側にある3bは画面制御部でホスト側から通知される入出力要求を制御するワークステーション側の制御プログラム、53はメニュー画面制御部でメニュー化された入力フィールドを制御するプログラム、54は情報ファイルでメニューに表示する内容を貯えた情報ファイルである。図26は図25の画面入出力情報ファイル4aの例を示す図であり、図において14はテキスト情報管理部でテキスト情報15と項目情報18を管理する。15はテキスト情報でテキスト制御情報24が含まれる。18は項目情報で項目の入出力区分、項目のタイプ、長さ等が格納されている。この画面入出力情報ファイル4aはホストコンピュータ側に存在する。図27は図25の情報ファイル54の例を示す図であり、図において14はテキスト情報管理部で、テキスト情報15と項目情報18を管理する。15はテキスト情報で、テキスト情報にはテキスト制御情報24が含まれる。図28は図26、図27のテキスト情報15の例を示す図であり、図において、24はテキスト制御情報で、テキスト表示領

域をクリアするか等の情報である。55は項目メニュー表示情報へのチェーンで項目メニュー表示情報56を指している。56は項目メニュー表示情報で入力フィールドをメニューに変えたときに表示させる入力データを格納している。

【0003】 次に動作について説明をする。端末側の動作を概略述べると、画面入出力時にはホスト計算機からのデータストリームを受け取り、簡単なメニュー情報が入った情報ファイル54を参照して、画面制御部3bが画面表示をする。情報ファイル54の内容も簡単であるが、画面制御自体も固定的なことしかできない。即ち、画面入出力には以下の4つの種別がある。1つ目は、項目で、カーソルが位置しているフィールドのデータのみを入力の対象とするもので、オペレータが入力したデータを1項目ずつ入力内容のエラーチェック、再入力処理等の木目細かな制御をしてアプリケーションを動作させるデータとして記憶する。2つ目は、テキスト入力で、入力、出力項目及び出力定数を含んだ集団であるテキスト単位で入力を行う。テキスト入力は1回のEnter（入力）キーの押下によりテキストに含まれる項目の入力ができる。端末からホストに渡される1回の通信処理にて複数の項目のデータを受け渡すことができるため入出力性能に優れる。3つ目は、項目出力で、一つの項目のみを画面上に出力するとき用いる。このときは出力定数の出力や項目属性（色情報）の出力はない。4つ目は、テキスト出力で、テキストに含まれる出力項目、出力定数、入力項目及び出力項目の項目属性（色、入出力区分等）の出力をまとめてテキスト単位で行う。図25において実行可能モジュール2aから項目入力要求がでると、実行制御部1bは画面制御部3bへ入力要求通知をする。画面制御部3bは実行制御部1bから渡された、図28の項目メニュー表示情報へのチェーン55のオフセットにより項目メニュー表示情報56を得て、メニューに入力候補となる入力データを表示する。また、実行可能モジュール2aからテキスト入力要求がでると、図26のテキスト情報管理部14のオフセットが渡されるので、画面制御部3bはテキストに含まれる項目分だけ項目入力と同様の動作をくり返し指定のテキスト分メニューを作り出す。

【0004】 上記構成によれば単純な項目入力、テキスト入力、項目出力とテキスト出力は可能であるが、それぞれのオペレータに見易くするための画面上の木目細かな制御、項目入力、テキスト入力のチェック等が固定であり、オペレータに優しい入出力制御はできなかった。勿論、ワークステーション側にインテリジェンスを持たせて、要求仕様に対応する操作性向上プログラムを作成すればよいが、各ワークステーション毎に、又はまとめてワークステーション用として開発したとしてもホスト側のアプリケーションプログラムと整合性をとらねばならずソフトウェア生産性を悪化させるだけでなくバージ

ョン変更や、各ワークステーション毎のプログラム管理等、保守性もよくない。

【0005】

【発明が解決しようとする課題】上記のように従来の方法では、端末は小規模であり高速のプロセッサ、大容量のメモリがなく、操作性の向上ができないという課題があった。近年のパーソナル計算機の発達により、このいわゆるパソコンを利用した端末はそれぞれパソコン上でプログラムすれば操作性の向上が可能であるが、そのためには端末側でのプログラミングが必要で、ホスト側との整合性を考慮しなければならず、保守管理も困難であるという課題があった。

【0006】本発明は上記の課題を解決するためになされたもので、既存のアプリケーション・プログラムを有効利用し、ソフトウェアの生産性と保守性を向上して、しかも端末での操作性を向上した端末操作性向上方法を得ることを目的とする。

【0007】

【課題を解決するための手段】この発明に係るアプリケーション・プログラム使用の端末操作性向上方法は、入出力を実行する際に参照する入出力情報ファイルの領域を拡張し、アプリケーション・プログラムに関するテキスト情報に付加情報を付加して拡張入出力情報ファイルを作成するステップと、拡張入出力情報ファイルをホスト計算機から端末へダウンロードするステップと、端末では拡張画面制御部を設け、拡張画面制御部が拡張入出力情報ファイルを参照して入出力データを出力またはメモリへ記憶するステップとを備えた。

【0008】また更に、拡張入出力情報ファイルは、テキスト情報を索引部とチェック情報を含む詳細情報とに分け、詳細情報部分のみに付加情報を追加するようにした。

【0009】

【作用】この発明によるアプリケーション・プログラム使用の端末操作性向上方法は、ホスト計算機側で一括製作された拡張画面入出力情報ファイルが端末へダウンロードされ、端末側ではこれに従ってデータがチェック、出力または入力処理され、あるいはエラー出力される。

【0010】また更に、拡張画面制御部の既存のロードモジュールが参照する索引部は変更されておらず、従って既存のアプリケーション・プログラムに基づいて動作する入出力は何の影響も受けない。

【0011】

【実施例】

実施例1. 以下、この発明の一実施例を図について説明する。図1はアプリケーション実行時の構成を示している。図において、ホストコンピュータ100側の2はアプリケーションプログラムの一部である実行可能モジュールで、1は実行可能モジュール2の入出力を制御する実行制御部である。一方、3はホストから通知される画

面入出力を制御する画面制御部であり、ワークステーション200上に存在する。同じく4bは新形式になった画面入出力情報ファイルで、5はワークステーションウィンドウ画面である。ホストとワークステーションはネットワーク300を介して接続されている。図2は図1のアプリケーション実行のための装置を作成するための開発時の構成を示した図である。図において、6はホスト側開発環境制御部、8はコンパイラ、9は画面入出力情報ファイルダウンロード制御部、10はアプリケーションプログラムソースで、4a従来形式の画面入出力情報ファイルである。一方、ワークステーション側にある7はワークステーション側開発環境制御部で、4bはホスト側からダウンロードされる画面入出力情報ファイルである。図3は画面入出力ファイルに新機能を付加するためのエディタの動作を説明するための図である。図において、12は画面入出力ファイルエディタ、13は画面編集用ワークステーションウィンドウ画面である。

【0012】図4は画面入出力情報ファイルの構造の例を示した図である。図において14はテキスト情報管理部で、画面入出力情報ファイルに含まれるテキストの数だけ存在する。15はテキスト情報であり、テキスト情報管理部14からポイントされる。16は野線情報であり、テキスト情報15からポイントされる。17はビットマップ情報であり、テキスト情報15からポイントされる。18は項目情報であり、テキスト情報管理部14からポイントされる。19は入力内容チェック情報であり、項目情報18からポイントされる。図5はテキスト情報管理部の構造を示した図である。図において、20はテキスト情報へのチェインである。画面入出力情報ファイルの数+1個が生成され、最後はこの部分が-1となり、テキスト管理部の終わりを示すストップとなる。21はテキスト名である。22はテキストに含まれる項目情報へのチェインであり、項目の数だけ存在する。テキストに含まれる項目の数+1個が生成され、最後はこの部分が-1となり、テキスト管理部の終わりを示すストップとなる。23は22で示したストップである。図6はテキスト情報の構造を示した図である。図において、24はテキスト制御情報、25は野線情報へのチェインであり、野線データがないときは-1となる。26はビットマップ情報へのチェインであり、ビットマップデータがないときは-1となる。

【0013】図7は野線情報の構造を示した図である。この情報は図6の野線情報25よりポイントされる。図において、27は野線データストリームで、野線表示データである。図8はビットマップ情報の構造を示した図である。この情報は野線情報25よりポイントされる。図において、28はビットマップデータで、イメージデータのビットマップである。図9は項目情報を示す図であり、図5の項目情報へのチェイン22からポイントされる。図において、31は入出力区分であり、項目フィ

ールドの入力、出力、入出力を区別する。32は項目のタイプであり、文字、数字、定数を区分する。32は項目のタイプであり、文字、数字、定数を区分する。33は項目の長さを示し、34は項目が数字タイプの場合の小数桁長を示し、35は項目フィールドの横（カラム方向）位置を示し、36は項目フィールドの縦（行方向）位置を示し、37は出力時の文字列編集パターンを示す。38は文字の色、サイズ、フォント、入力ガイダンスの有無、入力フィールド種別を示す属性であり、39は入力内容チェックへのポインタであり、40は属性38がリストボックス（メニュー）形式、ラジオボタンの時に、メニュー及びボタンの横に表示する文字列の数を示す。41は属性38がリストボックス（メニュー）形式、ラジオボタンの時に、メニュー及びボタンの横に表示する文字列データであり、複数のデータはNULLで区切られる。42は属性が入力ガイダンス有りの時の操作ガイダンスに表示するデータの内容である。

【0014】図10は項目入力時に、入力内容が正しいかをチェックするためのチェック情報の構造を示した図である。このチェック情報は図9の39からポイントされる。図において、43はチェック種別であり、日付検査、数字検査、値検査、負の数検査の種別を表す。44はチェックで入力エラーとなった場合のプログラムの処置の区分であり、メッセージボックスを出して、再入力する／アポートするを区分する。45はチェックでエラーとなった場合のメッセージボックスに出すメッセージデータである。46はチェック種別が値検査の時比較するデータの数を格納する。47はチェック種別43が値検査の時比較するデータであり、複数の時NULLで区切られる。図11は、画面入出力情報ファイルエディタ12が制御するワークステーションウィンドウ画面5の例である。図において、48は項目属性ダイアログ、49は項目属性変更ボタン、50は野線作成ボタン、51はイメージ作成ボタン、52は項目／定数移動ボタンである。以下、図12はテキスト出力時の動作フローを示す図、図13は項目出力時の動作フローを示す図、図14は項目入力の動作フローを示す図、図15はテキスト入力の動作フローを示す図、図16実行時エラーのメッセージボックス化のイメージを示す図、図17はメッセージボックス化の処理フローを示す図、図18はコンパイラの構成と流れを示す図である。

【0015】次に上記構成のシステムにおいて、操作性を向上した制御方法を説明する。実施例1においては実行可能モジュール2を再作成することなく画面インタフェースをよくする例を示す。画面インタフェースを向上させるために、画面入出力情報ファイル4bを4aから変更する。図2の画面入出力情報ファイルダウンロード制御部9とワークステーション側開発環境制御部7を用い、既存の画面入出力情報ファイル4aをワークステーション側に転送しかつ、ワークステーション側開発環境

制御部7により新形式の画面入出力情報ファイル4bを生成する。即ち、既存のアプリケーションプログラムを利用し、それらと整合性をとりつつ、しかもホスト計算機側で一括して端末ワークステーション側の操作性を向上して保守性を良くしようとする。具体的には画面入出力情報ファイル4a中の索引部（ヘッダ）はそのままにして、操作性に係わる画面インタフェース、特に詳細情報と、入出力データのチェック情報を付加または改良しようとするものである。これらの情報に基づいて機能強化された画面制御部が入出力制御を行う。以下に具体例に基づいて詳細を説明する。画面入出力情報ファイル4aと画面入出力情報ファイル4bとの違いは、図4と従来の図20との比較で判るように、テキスト情報15から野線情報16、ビットマップ情報17がチェーンされていること、および項目情報に入力内容チェック情報19へのチェーン、操作ガイダンスデータ、選択（メニュー表示）データが付加されている点である。索引部のテキスト情報管理部14の各要素のサイズは変わらない。即ち既存のアプリケーションプログラムは何の手を加える必要もない。テキスト情報15へのチェーン、項目情報へのチェーンの値が変わるだけである。ダウンロード時点では既存の情報のみが画面入出力情報ファイル4bに反映される。

【0016】図3の画面入出力ファイルエディタ12は、ホスト計算機からダウンロードされた画面入出力情報ファイル4bを読み込み、図11の画面編集用ワークステーションウィンドウ画面5に指定されたテキスト内容を表示する。まずウィンドウ画面上のボタン等と対応する情報の格納位置との対応を説明する。文字属性（文字サイズ／フォント／色、操作ガイダンスの追加、表示位置）を変更する場合は、図11の項目属性変更ボタン49を押すと項目属性ダイアログ48が画面に表れる。実はそれまでは画面にでていない。文字サイズ／フォント／色を変更する場合は項目属性ダイアログ48の属性の文字サイズ／文字フォント／色を変更する。画面入出力ファイルエディタ12は変更された情報を図9の項目情報の属性38に反映する。入力フィールドに対しては項目属性ダイアログ48の操作ガイダンスの設定ができる。操作ガイダンスは文字列であり、入力された文字列は操作ガイダンスデータ42に格納される。項目／定数の表示位置を変更する場合は項目属性ダイアログ48の横位置／縦位置を変更する。この変更した内容にて画面入出力情報ファイルエディタ12は横位置35、縦位置36を更新する。また、項目／定数移動ボタン52を押しマウスを用いて移動させることもできる。このときも変更した内容は横位置35、縦位置36に格納される。野線を付加するまたは、修正する場合は図11の野線ボタン50を押し、マウス操作により変更／生成する。画面入出力情報ファイルエディタは野線データストリーム27を更新／追加する。ビットマップデータ（イメージ

データ)を付加する場合は図11のイメージ作成ボタン51を押し予め標準のツールで作成したビットマップデータ11を読み込み図8ビットマップデータ28に格納する。

【0017】アプリケーションプログラムの操作性向上に関連する実行時の動作は以下の通りとなる。実行可能モジュール2からテキスト出力が出されたときは、図12のフローに示す動作を行う。実行可能モジュール2から画面入出力情報ファイル4bの名前とテキスト情報管理部22が格納されている画面入出力情報ファイル4bの先頭からのオフセット、出力データがwriteの要求の引数として、実行制御部1へ通知される(図12の58)。実行制御部1は、ネットワークを介して画面制御部3へwrite要求の引数を通知する(図12の60)。画面制御部3は画面入出力情報ファイル4bから指定されたテキスト情報管理部14を得る(図12の61)。画面制御部3はテキスト制御情報24を画面に出力する(図12の62、63)。テキスト制御情報3は、テキスト表示領域をクリア等の情報である。次に罫線情報へのチェーン25から罫線データストリーム27を得て罫線を画面に出力する(罫線情報が-1のときは罫線の出力はない)(図12の64)。次にビットマップチェーン26からビットマップデータ28を得てイメージを画面に出力する(ビットマップ情報が-1のときはビットマップデータの出力はない)(図12の65)。次に画面制御部3は項目情報へのチェーン22からテキストに含まれる項目の数だけ項目出力を繰り返す(図12の66、67)。全項目の出力が完了すると画面制御部3は出力完了を実行制御部1へ通知する(図12の68)。実行制御部1は出力完了を実行可能モジュール2に通知する(図12の69)。

【0018】実行可能モジュール2から項目出力要求が出されたときは、図13のフローに示す動作を行う。実行可能モジュール2から画面入出力情報ファイル4bの名前と項目情報へのチェーン22が格納されている画面入出力情報ファイル4bの先頭からのオフセット、出力データがwriteの要求の引数として、実行制御部1へ通知される(図13の72)。実行制御部1は、ネットワークを介して画面制御部3へwrite要求の引数を通知する(図13の73)。画面制御部3は指定された画面入出力情報ファイルから、指定されたオフセットの項目情報のチェーン22を得て、項目情報のチェーン22から図9項目情報を得る(図13の74)。項目情報の横位置35、縦位置36に出力時の編集パターン37に従い、属性38に格納されている画面入出力情報ファイルエディタ12にて更新された文字フォント、文字サイズ、色にてワークステーションウィンドウ画面5に項目内容を出力する(図13の75)。画面制御部3は実行制御部1へ出力完了を通知する(図13の76)。実行制御部1は実行可能モジュール2に出力完了を通知す

る(図13の77)。

【0019】実行可能モジュール2から項目入力が出されたときの動作を図14に示す。実行可能モジュール2から画面入出力情報ファイル4bの名前と項目情報へのチェーン22が格納されている画面入出力情報ファイル4bの先頭からのオフセット、データ受け取り領域がread要求の引数として、実行制御部1へ通知される(図14の80)。実行制御部1は、ネットワークを介して画面制御部3へread要求の引数を通知する(図14の81)。画面制御部3は指定された画面入出力情報ファイルから、指定されたオフセットの項目情報のチェーン22を得て、項目情報のチェーン22から図9項目情報を得る(図14の82)。図9の操作ガイダンスデータ42をワークステーションウィンドウ画面5に出力し、図9の項目情報の横位置35、縦位置36に入力要求を出す。このとき、操作ガイダンスデータ42があるときは操作ガイダンスをウィンドウに表示する(図14の83)。データの入力がなされると画面制御部3は入力データが正しいかを図10入力チェック情報のチェック種別43にて、入力データが日付として正しいか、数値かどうか、比較相手データ47との大小関係をチェックする(図14の84)。入力内容が不正であったとき画面入出力情報ファイルエディタ12について追加されたエラー時のメッセージ内容45をメッセージボックスを使って表示し、再入力へいく(図14の85)。テキスト入力でない場合は画面制御部3は実行制御部1へ入力データを通知する(図14の86)。実行制御部1は実行可能モジュール2へ入力データを通知する(図14の87)。

【0020】実行可能モジュール2からテキスト入力が出される場合は、図15のフローに示す動作を行う。実行可能モジュール2から画面入出力情報ファイル4bの名前とテキスト情報管理部22が格納されている画面入出力情報ファイル4bの先頭からのオフセット、データ受け取り領域がread要求の引数として、実行制御部1へ通知される(図15の90)。実行制御部1は、ネットワークを介して画面制御部3へread要求の引数を通知する(図15の91)。画面制御部3は画面入出力情報ファイル4bから指定されたテキスト情報管理部14を得る(図15の92)。次に画面制御部3は要求されたテキストに含まれている入力項目の一番初めに定義された入力フィールドへカーソルを位置付ける。アプリケーションの操作員はテキストに含まれる入力項目に対しデータを入力していく。Enterキーが押下されると、画面制御部3は項目情報へのチェーン22からテキストに含まれる項目の数だけ項目入力を繰り返す(図15の92~93)。全項目のデータの取得が完了すると画面制御部3は入力完了と入力データを実行制御部1へ通知する(図15の94)。実行制御部1は入力完了を実行可能モジュール2に通知する(図15の95)。

【0021】上記のテキスト出力、項目出力、項目入力、テキスト入力の動作に際して、画面入出力情報ファイル4bの4aからの内容変更により操作性向上のデータ変更がなされた場合の具体的な画面の例を示す。図16は表示フォントを121、122、123の部分のみを他と同一の大きさと形から変更した例である。これは図11のワークステーションウィンドウ画面5のカーソル表示位置で、項目属性ダイアログ48の属性を指定すれば変更ができる。つまり、図4の画面入出力情報ファイル4bの項目情報18の詳細である図9の属性38の領域を新たに追加することで実現できる。項目出力時、つまり図13の動作時にこの属性38を見て文字サイズ、文字フォントで表示するからである。同様に、図17はウィンドウ画面の例を見やすくして操作性を向上した例である。カーソル等を使って罫線126、127のように罫線位置を指定し、その線の種類等を指定すると、図4の画面入出力情報ファイル4bの付加した罫線情報16から図7の罫線データストリーム27が得られて、画面表示動作時に図12のステップ64で罫線が表示される。

【0022】図18は、ビットマップデータを付加してウィンドウ画面を見やすくして操作性を向上した他の具体的な例である。他からの図形を利用するかまたはマウス等を使ってパス図形131のビットマップを追加する。このデータは図4の画面入出力情報ファイル4bの付加したビットマップ情報17から図8のビットマップデータ28が得られて、画面表示動作時に図12のステップ65でパス図形が表示される。図19は、操作ガイダンスを付加して入力時の操作性を向上した例である。行き先コードのコード表示欄132を設け、地名コードを表示する設定ができる。従来のダム端末では、特に罫線とか出力領域を自由に設定した操作ガイダンスをつくることは難しかったが、本発明によれば入力フィールドに付随した設定が可能なので、かなり自由なガイダンスを作成できる。これら設定された値は画面情報ファイルエディタ12により、図9の項目情報に付随して格納される。そして項目入力時、図14の動作フローのステップ83で図9の操作ガイダンスデータ42の値を子ウィンドウに表示する。図20は、文字の表示位置を変更した例を示す画面である。この例では従来の24行×80列で固定の文字表示位置を、画面入出力情報ファイル4bの項目情報18の横位置35、縦位置36に格納し、表示動作時には対応する欄の表示位置をこの値で制御する。実施例1の方法によれば、ワークステーション側のプログラムをアプリケーション対応で構築する必要がなく、ソース管理、モジュール管理が容易で保守性、運用性が高い。更に、アプリケーションの動作はホスト側のみであるため、システム管理が容易である。

【0023】実施例2. チェック機能を強化して操作性を向上する例を説明する。従来は実行時にエラーが発生

した場合『コンソールに切り替えてください』等の表示がされるだけでエラーの詳細は直接見られず、まして詳しい内容は判らなかった。本実施例で実行時エラーが発生した場合の動作を図21に基づいて説明する。なお、図22はエラー発生時の表示画面の例であり、図23はテーブル指標がエラーになったときのエラー表示の例である。いずれもエラーメッセージボックス化をしてある。実行可能モジュール2内、又は実行制御部1内にてエラーが発生した場合は、実行制御部1内のエラー処理ルーチンへエラーが通知される。実行制御部1は画面制御部3へエラーを通知する。画面制御部3はワークステーションウィンドウ画面5に図23の実行時エラーのメッセージボックスを表示する。ここで、ユーザが終了するを選べば、ステップ108で画面制御部3は実行制御部1へ終了を通知する。実行制御部1は実行可能モジュール2を終了させる。また、ユーザがエラーをリカバリできるようなエラーの場合は、例えば図24で、ユーザが項目の値の変更を選択したとき、ステップ106で変更された値を実行制御部1に通知する。実行制御部1は実行可能モジュール2のデータ部内の内容を変更する。なお、実行時にユーザが修復不可能なエラーが発生した場合でも、図28に例を示すようにエラーコードまで表示して詳細が判るよう出力する。

【0024】実施例3. 実施例1では、実行可能モジュール2を再作成せずに操作性をよくしたが、アプリケーションソースプログラム10には一切の手を加えず再コンパイルすることによりデータ入力時にエラーが発生した場合のユーザインタフェースを向上させる手段について説明する。図24はコンパイラによる解析動作フローを示す図である。従来のコンパイラは図2の解析の流れの画面定義の解析ステップ115で、入力チェックをし、その情報を実行可能モジュール2のデータ部に格納する。従来は実行制御部1がそのデータよりチェックを行っていた。しかし、本発明においては、図24の画面定義部の解析ステップ115で、入力チェックをしたその情報は、図4の入力内容チェック情報19に格納する。実行時の動作について説明をする。図14は項目入力の際の動作であるが、ワークステーションウィンドウ画面5からデータが入力されると、画面制御部3は図10に示す入力内容チェック情報によってチェックする。データがチェックに引っかかったときは画面入出力情報ファイルエディタ12にて追加されたエラー時のメッセージをエラー時のメッセージ内容45から得てメッセージボックスに表示をする。ユーザがメッセージボックスを閉じると再入力にいきこれを繰り返す。

【0025】本実施例では、プログラムソースを再度コンパイルすることにより、端末側で処理可能な部分を抽出し、操作性の向上とホスト負荷分散を達成している。このために、図24に示す機能を強化したコンパイラと、実施例1で述べた入力内容チェックを制御する画面

制御部3を備えた。項目の入力に関して、値の範囲、漢字か、数字か、等をチェックすることがよくあるが、このチェック内容をプログラムソースが抽出し、このチェック内容を画面入出力情報ファイル4bへ格納し、端末で入力時にこのチェック内容を見て入力データの整合性をチェックする。その結果、ホスト側でチェックする必要がなくなる。また、端末側でチェックするので、入力エラー時には図10に格納された(画面入出力情報ファイルエディタ12にて変更、追加する)入力内容チェック情報により判り易いエラーメッセージボックスで表示する。

【0026】

【発明の効果】以上述べたようにこの発明によれば、アプリケーション・プログラムに関係するテキスト情報に付加情報を付加した拡張入出力情報ファイルを作成するステップと、これを端末へダウンロードするステップと、拡張画面制御部が拡張入出力情報ファイルを参照して入出力データを処理するステップとを備えたので、既存のアプリケーション・プログラムが有効利用でき、しかも端末での操作性を向上できる効果がある。

【0027】また更に、拡張入出力情報ファイルは、テキスト情報を索引部とチェック情報を含む詳細情報とに分けて詳細情報部分のみに付加情報を追加するようにしたので、既存のアプリケーション・プログラムには影響を与えず、プログラムの生産性と保守性を高く維持できる効果がある。

【図面の簡単な説明】

【図1】 この発明の実施例1の端末操作性向上方法を実行するシステム構成と動作フローを示す図である。

【図2】 実施例1の操作性向上を行うための開発時の構成とダウンロードのための構成を示す図である。

【図3】 実施例1の拡張画面入出力情報ファイルを作成するためのエディタの説明図である。

【図4】 実施例1の画面入出力情報ファイルの例を示した図である。

【図5】 実施例1のテキスト情報管理部の構造の例を示した図である。

【図6】 実施例1のテキスト情報の構造の例を示した図である。

【図7】 実施例1の野線情報の構造の例を示した図である。

【図8】 実施例1のビットマップ情報の構造の例を示した図である。

【図9】 実施例1の項目情報の構造の例を示した図である。

【図10】 実施例1の入力内容チェック情報の構造の例を示した図である。

【図11】 実施例1の画面入出力情報ファイルエディタが制御する画面の例を示す図である。

【図12】 実施例1の端末側のテキスト出力動作を示

すフロー図である。

【図13】 実施例1の端末の項目出力動作を示すフロー図である。

【図14】 実施例1の端末の項目入力動作を示すフロー図である。

【図15】 実施例1の端末のテキスト入力動作を示すフロー図である。

【図16】 実施例1で表示フォントを変更した画面を示す図である。

【図17】 実施例1で野線を追加した画面を示す図である。

【図18】 実施例1でビットマップデータを追加した画面を示す図である。

【図19】 実施例1で操作ガイダンスを追加した画面を示す図である。

【図20】 実施例1で文字の表示位置を変更した画面を示す図である。

【図21】 実施例2の端末が実行時エラーの際にメッセージボックス表示をする動作フローを示す図である。

【図22】 実施例2の端末が実行時エラー発生の際に表示する画面の例を示す図である。

【図23】 実施例2の端末の実行時エラー発生の際のメッセージボックス表示例を示す図である。

【図24】 コンパイラによる解析の流れを示す図である。

【図25】 従来の画面入出力制御方法を実行する装置構成を示す図である。

【図26】 従来の画面入出力情報ファイルの構成の例を示す図である。

【図27】 従来の情報ファイルの構造の例を示す図である。

【図28】 従来のテキスト情報の構造の例を示す図である。

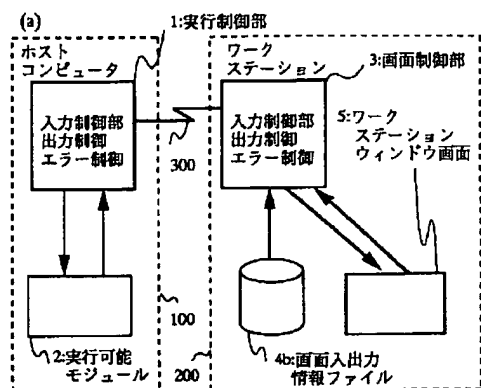
【符号の説明】

1 実行制御部、2 実行可能モジュール、3 画面制御部、4a 画面入出力情報ファイル、4b 画面入出力情報ファイル、5 ワークステーションウィンドウ画面、6 ホスト側開発環境制御部、7 ワークステーション側開発環境制御部、8 コンパイラ、9 画面入出力情報ファイルダウンロード制御部、10 アプリケーションプログラムソース、11 ビットマップデータ、12 画面入出力情報ファイルエディタ、14 テキスト情報管理部、15 テキスト情報、16 野線情報、17 ビットマップ情報、18 項目情報、19 入力内容チェック情報、20 テキスト情報へのチェイン、21 テキスト名、22 項目情報へのチェイン、23 ストップ、24 テキスト制御情報、25 野線情報へのチェイン、26 ビットマップへのチェイン、27 野線データストリーム、28 ビットマップデータ、31 入出力区分、32 項目のタイプ、33 項目の長

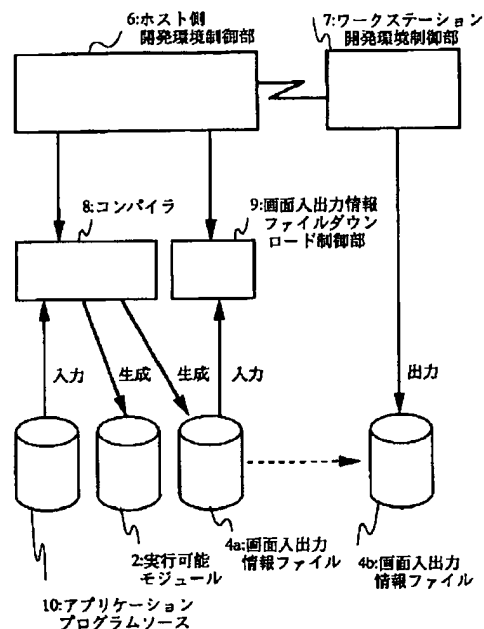
さ、34 項目の小数桁、35 横位置、36 縦位置、37 出力時の編集パターン、38 属性、39 入出力内容チェックへのポインタ、40 選択データの数、41 選択データ、42 操作ガイダンスデータ、43 チェック種別、44 エラー時の対応種別、45 エラー時のメッセージ内容、46 比較相手の数、47

比較相手データ、48 項目属性ダイアログ、49 項目属性変更ボタン、50 野線作成ボタン、51 イメージ作成ボタン、52 項目/定数移動ボタン、53 メニュー画面制御部、54 情報ファイル、55 メニュー項目情報へのチェーン、56 メニュー表示情報。

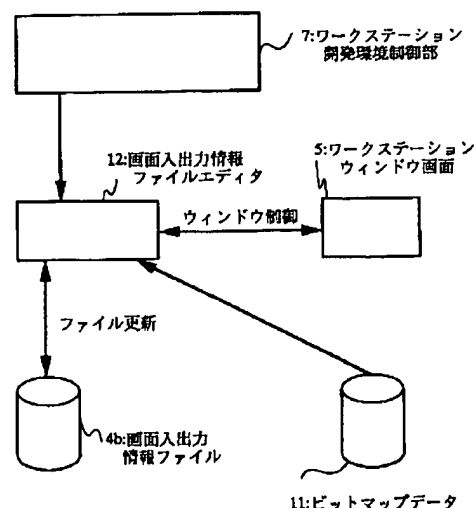
【図1】



【図2】



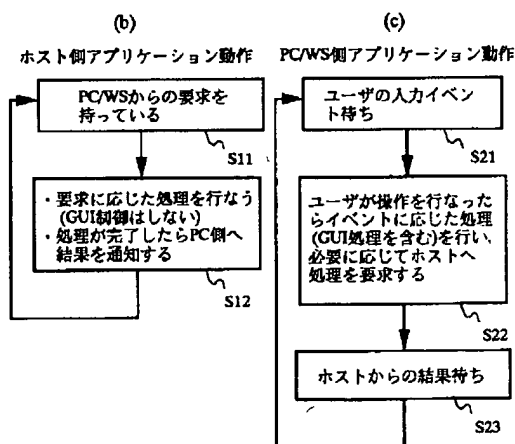
【図3】



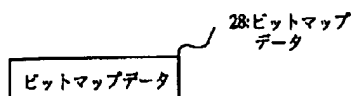
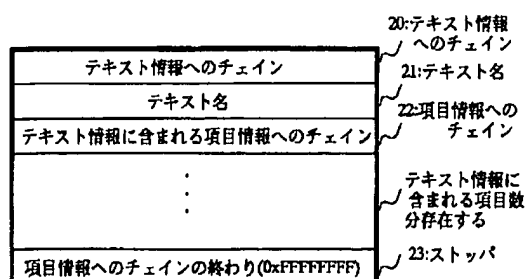
【図7】



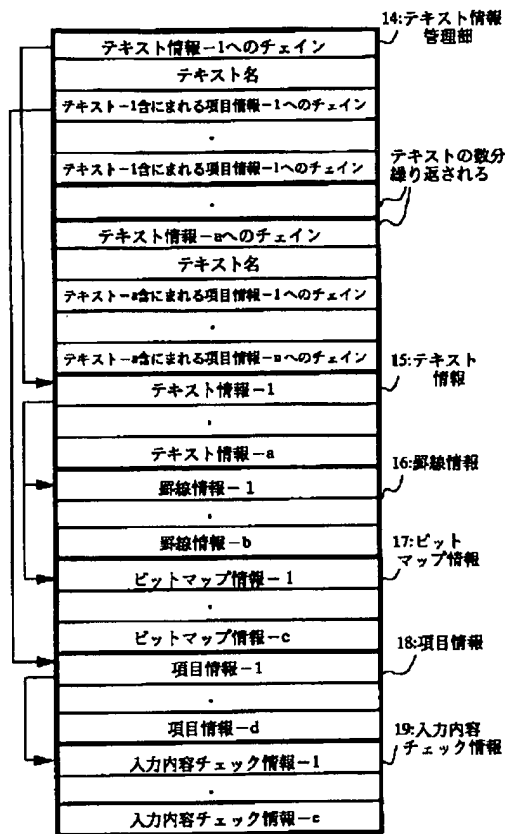
【図5】



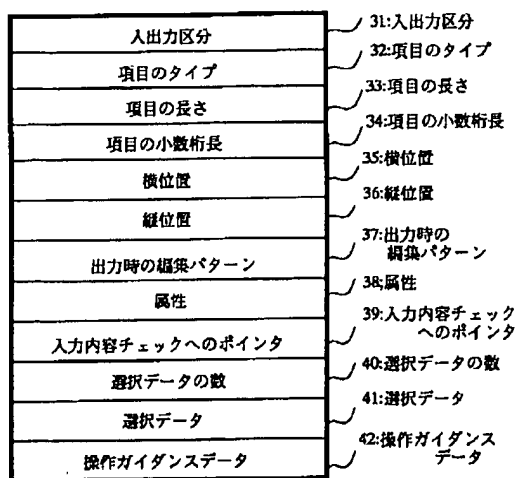
【図8】



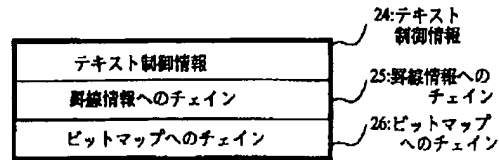
【図4】



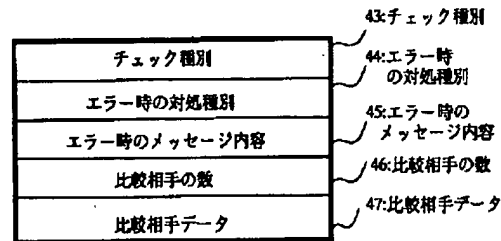
【図9】



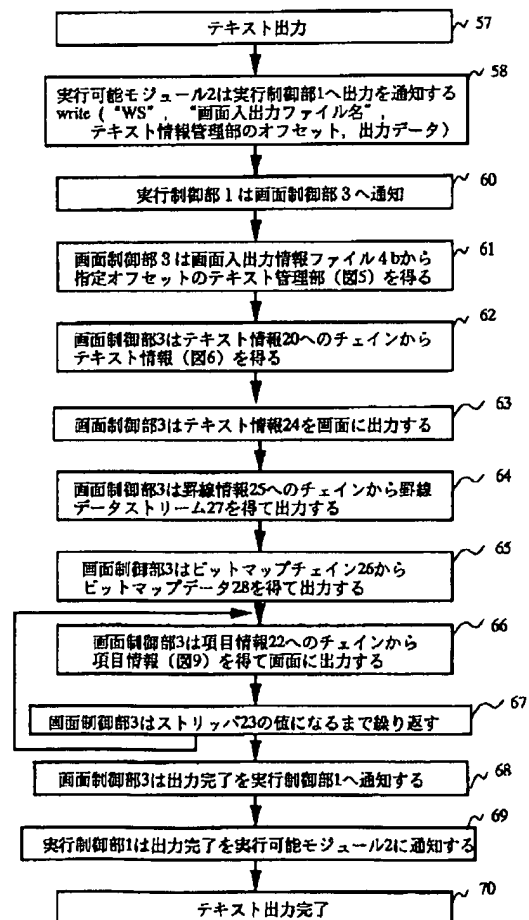
【図6】



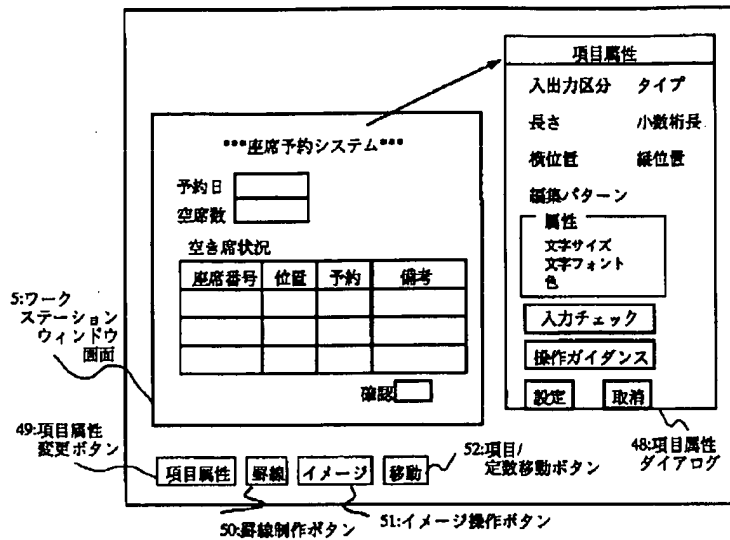
【図10】



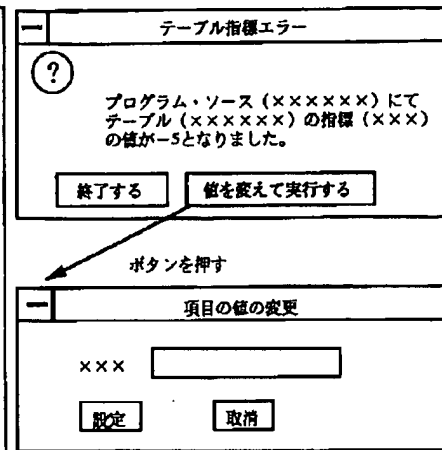
【図12】



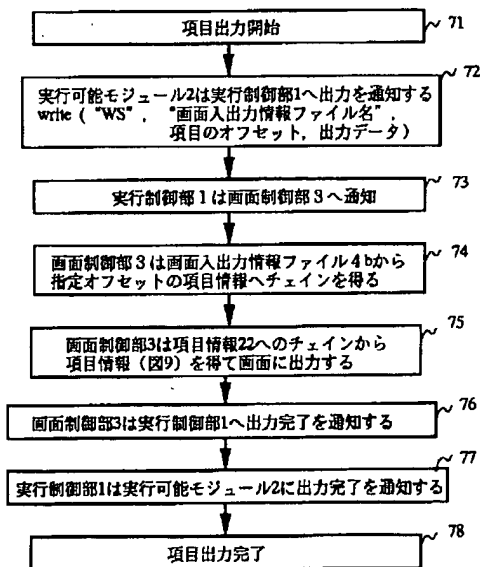
【図 1 1】



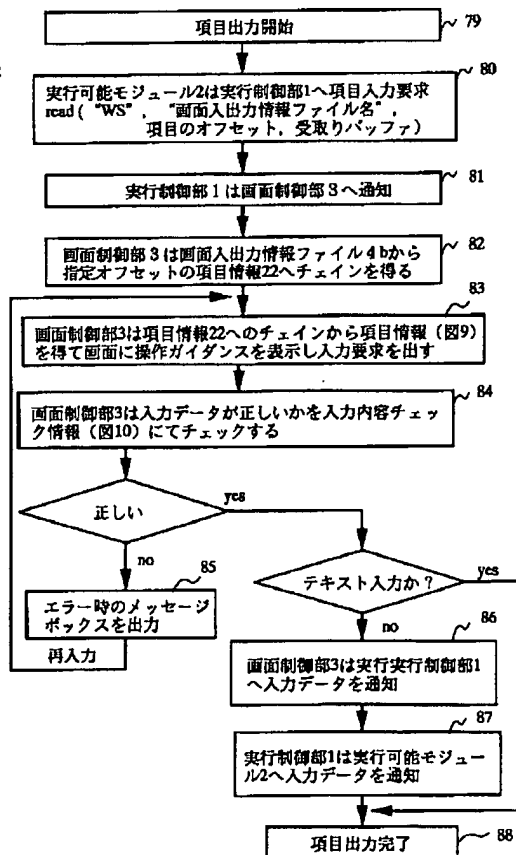
【図 2 3】



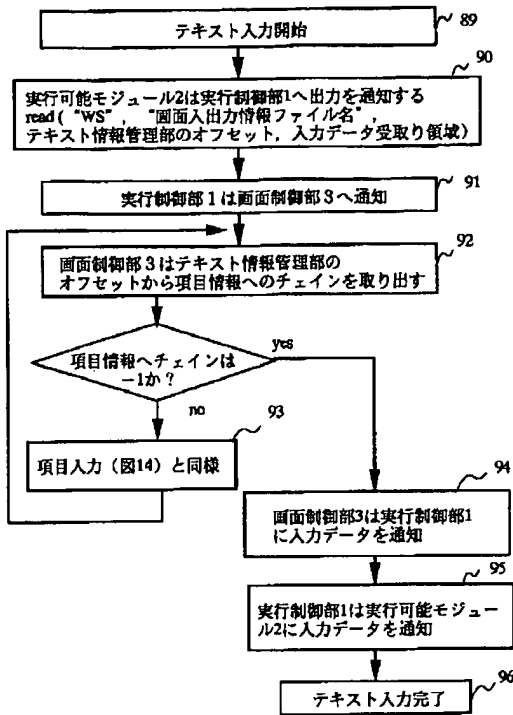
【図 1 3】



【図 1 4】



【図15】



【図17】

*** 座席予約システム *** 99年99月99日

予約日 999999

行き先コード 999999

乗車地コード 999999 下車地コード 999999

*** 空き席状況 ***

バスコード	発車時間	空席数

*** 予約 ***

バスコード 999999 予約席数 99

終了 前頁後頁

F0 F1 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

【図16】

*** 座席予約システム *** 99年99月99日

予約日 999999

行き先コード 999999

乗車地コード 999999 下車地コード 999999

*** 空き席状況 ***

バスコード	発車時間	空席数

*** 予約 ***

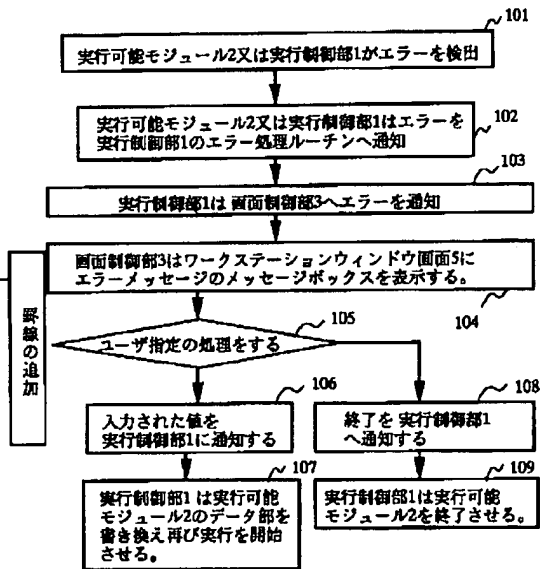
バスコード 999999 予約席数 99

終了 前頁後頁

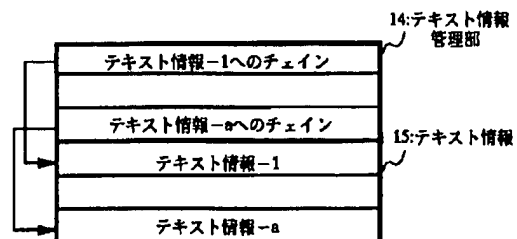
F0 F1 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

フォントを変更した

【図21】



【図27】



【図18】

座席予約システム 99年99月99日

予約日 999999
 行き先コード 999999
 乗車地コード 999999 下車地コード 999999

空き席状況

バスコード	発車時間	空席数

131

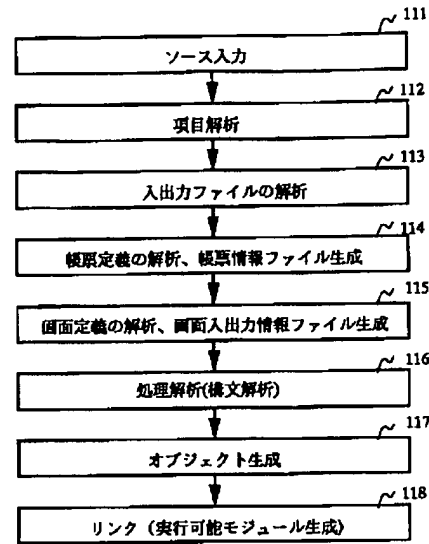
予約

バスコード 999999 予約席数 99

終了 前頁後頁
 F0 F1 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

ビットマップを追加

【図24】



【図19】

座席予約システム 99年99月99日

予約日 999999
 行き先コード 999999
 乗車地コード 999999 下車地コード 999999

空き席状況

バスコード	発車時間	空席数

132

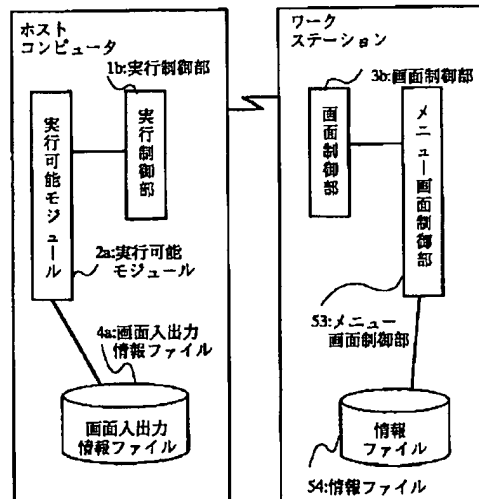
予約

バスコード 999999 予約席数 99

終了 前頁後頁
 F0 F1 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15

操作ガイダンスを追加

【図25】



【図22】

座席予約システム

実行時エラー発生

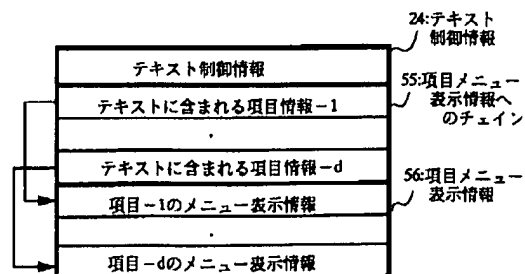
座席予約システムにてエラーが発生しました。

エラーコードは、0001です。

座席予約システムを終了します。

了解

【図28】



【図20】

*** 座席予約システム ***

99年99月99日

予約日 ~ 133

行き先コード

乗車地コード 下車地コード

*** 空席状況 ***

バスコード	発車時間	空席数

*** 予約 ***

バスコード 予約席数

134

終了 F1 F3 F4 F5 F6 F9 F10 F11 F12 F13 F14 F15

前頁後頁

【図26】

